Ty Limpasuvan

ACM 104 Homework 6

Problems 1, 2, 4, and 5 are attached.

(3) $F = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$

$$\det \begin{bmatrix} 1-\lambda & 1 \\ 1 & -\lambda \end{bmatrix} = 0 \rightsquigarrow -\lambda(1-\lambda)-1=0$$
$$\lambda^2-\lambda-1=0$$

Using the quadratic formula gives us:

$$\lambda = \frac{1 \pm \sqrt{5}}{2}$$

To find the corresponding eigenvectors, we solve:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \end{bmatrix}$$

Which gives: $\begin{array}{l} x+y=\lambda x \\ x=\lambda y \end{array} \rightarrow x = \frac{1\pm\sqrt{5}}{2} y$

$$V_1 = \begin{bmatrix} \frac{1+\sqrt{5}}{2} \\ 1 \end{bmatrix} \quad \text{and} \quad V_2 = \begin{bmatrix} \frac{1-\sqrt{5}}{2} \\ 1 \end{bmatrix}$$

So we can diagonalize F now:

$$F = \begin{bmatrix} \frac{1+\sqrt{5}}{2} & \frac{1-\sqrt{5}}{2} \\ 1 & 1 \end{bmatrix}\begin{bmatrix} \frac{1+\sqrt{5}}{2} & 0 \\ 0 & \frac{1-\sqrt{5}}{2} \end{bmatrix}\begin{bmatrix} \frac{1+\sqrt{5}}{2} & \frac{1-\sqrt{5}}{2} \\ 1 & 1 \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} \frac{1+\sqrt{5}}{2} & \frac{1-\sqrt{5}}{2} \\ 1 & 1 \end{bmatrix}\begin{bmatrix} \frac{1+\sqrt{5}}{2} & 0 \\ 0 & \frac{1-\sqrt{5}}{2} \end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt{5}} & -\frac{1-\sqrt{5}}{2\sqrt{5}} \\ -\frac{1}{\sqrt{5}} & \frac{1+\sqrt{5}}{2\sqrt{5}} \end{bmatrix}$$

```
% Problem 1
function [ e, v ] = ps6_1_TyLimpasuvan( A )
    [~, n] = size(A);
    x = rand(n, 1);
    threshold = 0.00001;

    while true
        temp = A * x;
        temp = temp / norm(temp, Inf);
        if (norm(temp - x, Inf) < threshold)
            x = temp;
            break;
        end
        x = temp;
    end
    ev = A*x;
    e = ev(1) / x(1);
    v = x / norm(x);
end
```

```
Not enough input arguments.

Error in ps6_1_TyLimpasuvan (line 3)
    [~, n] = size(A);
```

Published with MATLAB® R2016a

```
% Problem 2
load USAirTransportation;
[m, n] = size(A);
L = zeros(m, n);

% Converting A into a column-stochastic matrix
for j = 1:n
    aCol = A(:, j);
    L(:, j) = aCol / sum(aCol);
end
S = zeros(m,n);
S(:) = 1/n;
alphas = [0.1, 0.15, 0.2]';

% ranking vectors for different alpha values
for i = 1:size(alphas)
    Lb = (1 - alphas(i)) * L + alphas(i) * S;
    % Find the ranking vector
    [e, v] = ps6_1_TyLimpasuvan(Lb);

    % Find top 10 airports
    sortedV = sort(v, 'descend');
    maxValues = sortedV(1:10, 1);
    ids = zeros(0, 1);
    for j = 1:10
        idx = find(v == maxValues(j, 1));
        ids = [ids; idx];
    end

    fprintf('IDs of top 10 important airports for alpha = %.2f,\n',
 alphas(i));
    fprintf('descending:\n');
    for j = 1:size(ids)
        fprintf('%i  ', ids(j));
    end
    fprintf('\n\n');
end
```

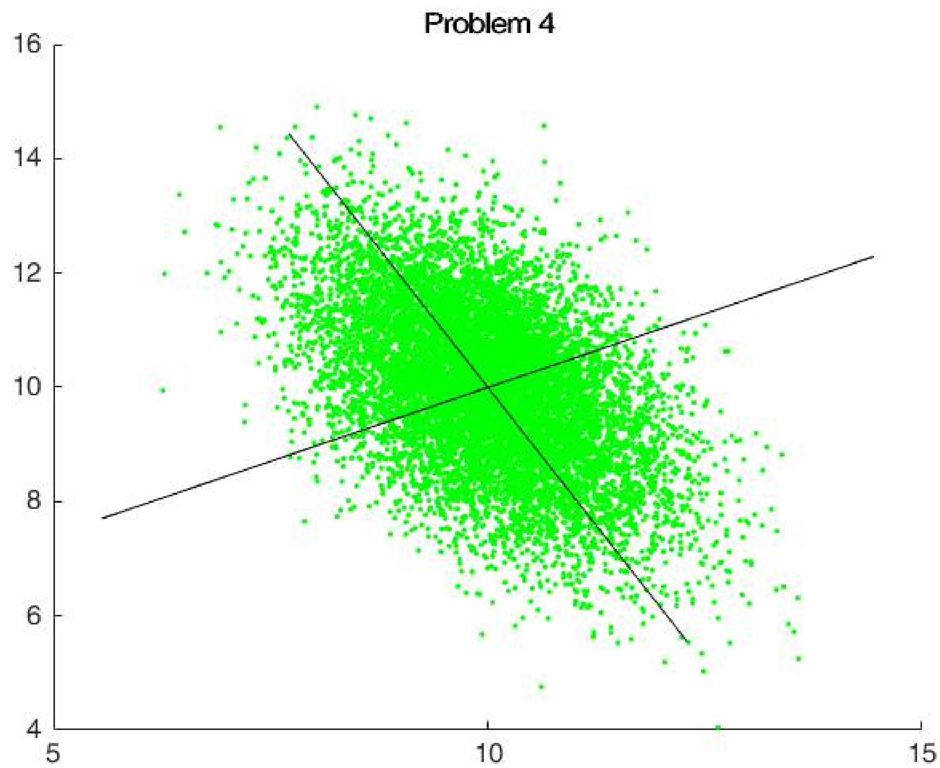
```
IDs of top 10 important airports for alpha = 0.10,
descending:
6  1  7  3  2  21  11  8  18  10

IDs of top 10 important airports for alpha = 0.15,
descending:
6  7  3  1  2  21  11  8  10  18

IDs of top 10 important airports for alpha = 0.20,
descending:
6  7  3  1  2  21  11  10  8  18
```

Published with MATLAB® R2016a

```
% Problem 4
load pca_data;
[m, n] = size(X);

% Plotting the original data
scatter(X(1, :), X(2, :), '.g');
hold on;
title('Problem 4');

% Part A
centroid = mean(X, 2);
Xm0 = zeros(m, n);
Xm0(1, :) = X(1, :) - centroid(1);
Xm0(2, :) = X(2, :) - centroid(2);
fprintf('Covariance matrix of X:\n');
C = 1/n * Xm0 * (Xm0')

% Part B
[V, D] = eig(C);
syms t
x = centroid(1) + t * V(1, 1);
y = centroid(2) + t * V(2, 1);
fplot(x, y, '-k');
x = centroid(1) + t * V(1, 2);
y = centroid(2) + t * V(2, 2);
fplot(x, y, '-k');

% Part C
Y = inv(V) * X;
centroidY = mean(Y, 2);
Ym0 = zeros(m, n);
Ym0(1, :) = Y(1, :) - centroidY(1);
Ym0(2, :) = Y(2, :) - centroidY(2);
fprintf('Covariance matrix of Y:\n');
Cy = 1/n * Ym0 * (Ym0')

% Part D
Vm = pca(X');
fprintf('Matlab pca():\n');
disp(Vm)
fprintf('My components:\n');
disp(V)
```

```
Covariance matrix of X:

C =

    0.9829   -0.6808
   -0.6808    1.9508

Covariance matrix of Y:
```

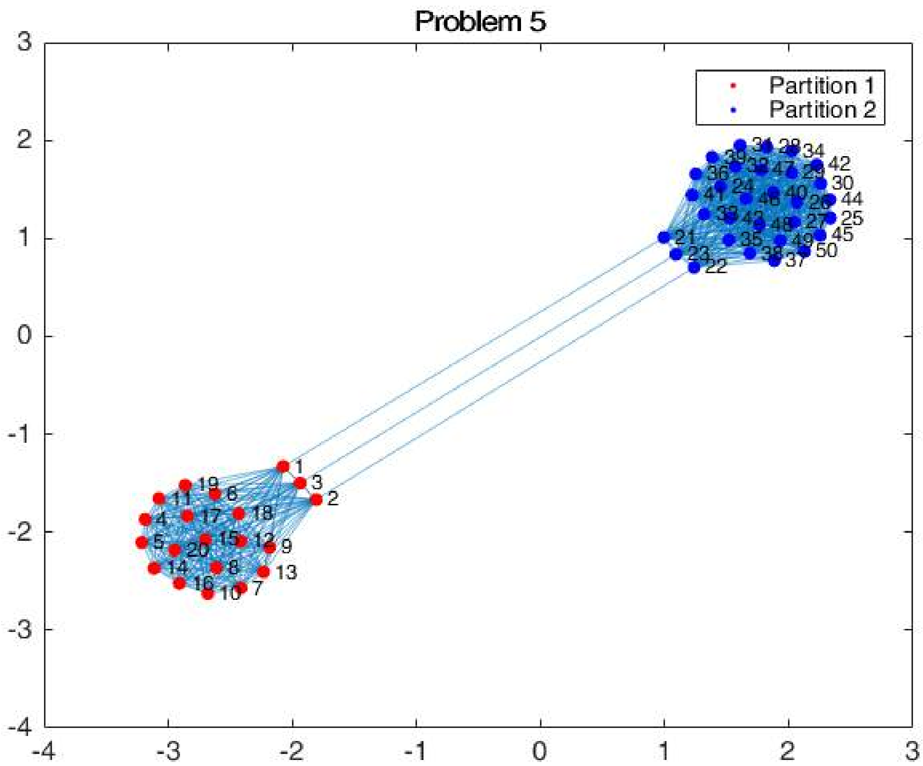
```
Cy =

    0.6316   -0.0000
   -0.0000    2.3021

Matlab pca():
   -0.4586    0.8886
    0.8886    0.4586

My components:
   -0.8886   -0.4586
   -0.4586    0.8886
```



Published with MATLAB® R2016a

```
% Problem 5
% Constructing adjacency matrix A and graph G
n = 50;
n1 = 20;
n2 = 30;
A = zeros(n);
A(1:n1, 1:n1) = 1;
A(n1+1:n, n1+1:n) = 1;
A(n1+1:n1+3, 1:3) = eye(3);
A(1:3, n1+1:n1+3) = eye(3);
A = A - eye(n);
G = graph(A);

% Plot result
h = plot(G, 'NodeColor', 'b');
hold on;
title('Problem 5');

% Construct graph Laplacian
Dv = degree(G);
D = zeros(n);
for i = 1:n
    D(i, i) = Dv(i);
end
L = D - A;

% Find 2 potential solutions
a1 = (n1 - n2)/sqrt(n);
a2 = 2*sqrt(n1*n2/n);
[V,D] = eigs(L, 2, 'SA');
v2 = V(:, 2);
splus = a1 * ones(n, 1) + a2 * v2;
sminus = a1 * ones(n, 1) - a2 * v2;

sortedSPlus = sort(splus, 'descend');
sortedSMinus = sort(sminus, 'ascend');
maxValues = sortedSPlus(1:n1, 1);
minValues = sortedSMinus(1:n1, 1);
maxIds = zeros(0, 1);
minIds = zeros(0, 1);
for j = 1:n1
    if length(maxIds) < n1
        idMax = find(splus == maxValues(j, 1));
        if ~ismember(idMax, maxIds)
            maxIds = [maxIds; idMax];
        end
    else
        maxIds = maxIds(1:n1, 1);
    end
    if length(minIds) < n1
        idMin = find(sminus == minValues(j, 1));
        if ~ismember(idMin, minIds)
```

```
                minIds = [minIds; idMin];
            end
        else
            minIds = minIds(1:n1, 1);
        end
    end
    for i = 1:n
        if ismember(i, maxIds)
            splus(i) = 1;
        else
            splus(i) = -1;
        end
        if ismember(i, minIds)
            sminus(i) = 1;
        else
            sminus(i) = -1;
        end
    end

    % Calculate the cut size and pick solution with the smallest
    splusCutSize = 1/4 * splus' * L * splus;
    sminusCutSize = 1/4 * sminus' * L * sminus;
    syms solution;
    if splusCutSize < sminusCutSize
        solution = splus;
    else
        solution = sminus;
    end

    % Highlight the relevant vertices on the graph
    ids = find(solution == 1);
    highlight(h, ids, 'NodeColor', 'r');

    h = zeros(2, 1);
    h(1) = plot(NaN,NaN,'.r');
    h(2) = plot(NaN,NaN,'.b');
    legend(h, 'Partition 1','Partition 2');
```



Published with MATLAB® R2016a